LESSONS LEARNED:
AN EXAMINATION OF CRYPTOGRAPHIC SECURITY SERVICES IN A FEDERAL AUTOMATED INFORMATION SYSTEM

Jim Foti
Donna Dodson
Sharon Keller
NIST, Building 820, Room 414, Gaithersburg, MD 20899

1. Introduction

Working with other agencies, the National Institute of Standards and Technology (NIST) recently completed a review of the security services implemented in an automated information system. During the review, several implementation flaws of the cryptographic security services were discovered which created vulnerabilities in an otherwise robust system. Based on findings during the review, recommendations were described and implemented to correct the security flaws and enhance the information system security already implemented in the system. The concerns described in the review could easily occur in other applications. Likewise, the recommendations provided by this review could be used by others to address security issues in other automated applications and systems.

1.1 Description of the System

To process information more efficiently, an agency recently developed an automated information system to replace and update its paper-based processing of work requests and approvals, in addition to the accounting associated with those requests. The system is based on a large relational database where electronic forms and user-provided data are stored in centrally located UNIX mainframes. A Wide Area Network is used to transmit information between users, and from the mainframes to PCs, so that users can manipulate and view the data and perform cryptographic security functions. There are currently 5000 system users, and it is projected that there will be 40,000 users by the end of 1997. Although the majority of the users are located in the United States, there are several sites in other parts of the world.

1.2 Use of Cryptography

Like many administrative applications, a replacement for handwritten signatures was required to totally automate this system. The agency also identified requirements for authentication and confidentiality; cryptography was employed to provide these security services. Because public key cryptographic standards were not available to the government during the design of the system, integrity, authentication, confidentiality, electronic signatures, and key management services were based on secret-key cryptography.

Key management is an essential component of the system, because it provides the foundation necessary to securely generate, store, distribute and translate keys. One of the fundamental principles for protecting keys is the practice of split knowledge and dual control. As defined in

ANSI X9.17-1985, *Financial Institution Key Management (Wholesale)* [1], split knowledge is "a condition under which two or more parties separately have key components which, individually, convey no knowledge of the resultant cryptographic key. The resultant key exists only within secure equipment". Dual control is explained in the standard as "a process of utilizing two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information." Split knowledge and dual control were implemented in the system to protect the central storage of user keys, secure the distribution of user tokens, and initialize all cryptomodules in the system to "authorize" their use in performing cryptographic functions within the system.

Central sites also play an important role in key management. ANSI X9.17 relies on Key Management Facilities and Key Translation Centers to manage secret keys and translate those secret keys for decryption and signature verification. In public-key systems, central sites typically include a Certification Authority (CA), which is an entity that issues and revokes public key certificates, and may even generate key pairs. In either case, whether in a secret- or public-key system, the security of the central sites is critical to the overall cryptographic security of the system.

2. Findings and Recommendations

The recommendations listed below address the successful addition of cryptographic security to an automated information system. They are based on the review of the system described in section 1, as well as experience gained by NIST in other cryptography-related activities. These recommendations could be applied to many systems implementing cryptographic security services, whether the type of cryptography being used is secret- or public-key based.

2.1 General System Recommendations

- *Have cryptographic modules tested before distributing them throughout the system.*

  Cryptographic services are provided using cryptographic modules (cryptomodules), which may include capabilities such as signature generation and verification (possibly involving key notarization), encryption and decryption, key generation, key distribution, etc. Examples of cryptomodules are smartcards, PCMCIA cards, PC adapters, and software modules, among other possible hardware, software, or firmware implementations.

  If a large number of cryptomodules are needed to provide security services in a system, then an undetected error in a cryptomodule's design could potentially affect the performance of a cryptographic function for every user in the system. For example, key notarization (for secret keys) might be done improperly by a cryptomodule, or verifications of a chain of public key certificates might not function correctly. Key notarization helps ensure that no party other than the signer of the data can use the data key to sign or encrypt information. Likewise, verifying a chain of public key certificates helps a signature verifier determine if a signature was generated with a particular key. If either of these functions were to be implemented incorrectly in a cryptomodule, the

potential for the dissemination of weak cryptography could be introduced into the system, possibly allowing for signature forgery or the verification of invalid signatures.

This shows the importance of testing a cryptomodule before using it to provide cryptographic security services in a large system. Currently, Federal agencies are required to procure cryptomodules which have either been validated under the Cryptographic Module Validation (CMV) Program or submitted to an accredited laboratory for CMV testing. A series of tests are run on cryptomodules to test for conformance to FIPS PUB 140-1, *Security Requirements for Cryptographic Modules* [2]. These tests encompass features such as physical and operating system security, roles and services, and others. Under the CMV testing, cryptographic algorithms are tested for conformance to standards such as the Data Encryption Standard [3], Digital Signature Standard [4], and Secure Hash Standard [5]. The algorithms are exercised to detect implementation flaws, by performing tests which compare results generated by the implementation against known values and values generated by a reference implementation. Such testing would help detect implementation flaws in a cryptomodule's design.

- *Make use of cryptographic services as much as possible.*

  By consistently replacing traditional methods of secure operation with cryptographic methods, the security and efficiency of a system improves dramatically. Benefits from implementing electronic or digital signatures include reducing the possibility of forgery, reducing processing time, and decreasing the burden of maintaining "traditional" paperwork. A system implementing cryptography will naturally generate new documentation, and the cryptographic technology should be applied in handling that documentation. Security officers, for example, may have to generate and sign requests for keys or cryptographic modules. Instead of using paper forms, electronic forms could be generated, signed, and sent to the appropriate parties, who can verify the signatures and act on the request in a very timely manner.

- *Provide consistent documentation and training to all system users, and place emphasis on educating them about cryptography.*

  It is particularly important that all users understand the system they are using, and they should be aware of their responsibilities and the procedures they must follow in ordinary as well as unusual circumstances. These procedures should be standard among all sites in the system. Of special importance are the central sites, where security officers are responsible for equipment that might generate and manage keys for system users. If no standard set of procedures is followed, weaknesses may be introduces into the system.

2.2 General Key Management

- *Cryptographic keys may need special physical protection.*

If keys or key components are stored on a token (e.g., floppy disk, PCMCIA, smartcard, etc.), this token may have to be stored in a special manner to prevent unauthorized individuals from accessing the key or key component. For example, if key components for starting a Certification Authority or Key Management Facility are stored on tokens which are secured in a safe, multiple people might have access to this token. Therefore, additional protection is needed for each token, possibly by using a tamper-evident envelope, to enable the token's owner to determine if a token was used by another person.

- *Make sure that users are aware of their liabilities and responsibilities, and that they understand the importance of keeping their keys secure.*

  The security of cryptographic keys in an electronic or digital signature system is the foundation of a secure system, therefore users *must* maintain control of their keys! Users must be provided with a list of responsibilities and liabilities, and each user should sign a statement acknowledging these concerns before receiving a key (if it is a long-term, user-controlled key). If different user types (e.g., security officer, regular user) are implemented in a system, they should be aware of their unique responsibilities, especially regarding the significance of a key compromise or loss.

- *Timeout features are important for protecting keys from compromise or misuse.*

  A timeout feature for a cryptographic module or token is important, to minimize the possibility of an unauthorized individual accessing an "active" cryptomodule and using it cryptographic keys. This could happen if a cryptomodule is left unattended by a user who has authenticated to it and loaded his cryptographic keys. One alternative is to force a user to periodically re-authenticate herself to a cryptomodule, rather than allow her to stay logged in for an indefinite amount of time. For sensitive applications, it may be necessary to restrict the hours during which they can take place.

2.3 Key Management Facility / Certification Authority

- *Maintaining control of central or root keys from the time of generation is critical.*

  Central or root keys are most likely to be used in sensitive applications such as encrypting user keys, signing a central key database for integrity, binding a key pair to a user, or generating user keys. If these keys are compromised, a complete system compromise becomes a very real threat. It is essential to maintain the security of these central keys from the very beginning - the generation process. No one but the proper owner(s) of a key or key component should ever be able to use that key or key component. If split knowledge and dual control are a requirement for central or root keys, then a failure to maintain split knowledge and dual control of those keys at *any* time in their lifecycle could present both a security problem and a potential system compromise.

- *Keep a log of when root keys are used.*

A record should be maintained of every instance that a central/root key is used. This should be an automated feature that is built into the system.

- *Sign all centrally stored data and encrypt sensitive data, such as secret and private keys.*

  All centrally stored data that is related to user keys should be signed for integrity, and possibly encrypted for confidentiality (all secret and private keys should be encrypted). Individual key records in a database - as well as the entire database - should be signed. To enable tamper detection, each individual key record should be signed, so that its integrity can be checked before allowing that key to be used in a cryptographic function. When signing the entire database, at least the important fields that do not change regularly should be signed (this allows for faster verification).

- *Prepare for the possibility of compromise!*

  It is imperative to have a contingency plan for the compromise or suspected compromise of central/root keys or key components at a central site; this should be established before the system goes "live". The contingency plan should address what actions should be taken with system software and hardware, central/root keys, user keys, previously-generated signatures, encrypted data, etc.

- *Sign and verify the code that implements the cryptographic functions.*

  Software at the central key management site should be electronically signed and periodically verified to check the integrity of the code. This provides a means of detecting the unauthorized modification of system software. Within a cryptomodule, this feature of generating and verifying a cryptographic checksum is required by FIPS PUB 140-1.

- *A system implemented for a Government agency should have its centrally stored keys and system software controlled by Government employees.*

  Proper control of central/root keys and key management software and hardware is critical to the security of the system. Federal employees should be in control of this material for a system operated for the Federal government. Once the system goes live, *unlimited* access to central data, code, and cryptomodules should *not* be given to non-government employees, including those who were contracted to develop and/or maintain the system. It is understood, though, that the agency may need outside assistance in maintaining the system.

- *Use different types of central and root keys, where possible, to maximize the scalability of the system and the integrity of cryptographic data.*

  Different “types” of root keys might be implemented to 1) bring up a new system, 2) initialize a new central site, or 3) serve as backup keys for the same central site. It is very important to have backup copies of central/root keys, since the compromise or loss

of those components could prevent access to keys in the central database, and possibly deny system users the ability to decrypt data or perform signature verifications.

- *Be aware of security issues when migrating from a prototype to a live system.*

  When moving the system from a prototype to a live phase, the safest strategy is to generate new central/root keys and reissue keys for other system users. However, if it is not feasible to do this, then prior to migration a review of the generation, distribution, and storage procedures used for the root keys should be performed, to ensure that their security was maintained throughout their lifecycle. Otherwise, a security flaw or compromise in the prototype phase could be passed on to the live system.

- *Keep the KMF/CA flexible for scalability*

  Allow for the possibility of multiple "central" sites. More than the original number may be required if more users are added to the system. Ramifications on the root keys should be considered, including 1) how are they stored, 2) how are root keys to be generated for and distributed to the new central site, and 3) how will database information be communicated to the new central site and used by holders of the new root keys.

2.4 Key Distribution

- *If a key is stored on a token, and a PIN is used to access the token, then* only *that token's owner should* ever *have possession of both the token and its corresponding PIN.*

  This applies to root security officers who may generate a token and its PIN, as well as any intermediaries. To prevent a courier from having sole control of both items, security officers should distribute the token and PIN in separate mailings (in separate packages mailed on different days). Receipt of each item should always be confirmed to the original sender. A failure to maintain control of this token and PIN could lead to a key compromise and the misuse of cryptographic functions within the system.

2.5 Key Storage and Destruction

- *Determine reasonable lifetimes for keys associated with different types of users.*

  Users with different roles in the system should have keys with lifetimes that take into account the users' roles and responsibilities, the applications for which the keys are used, and the security services which are provided by the keys (user/data authentication, confidentiality, data integrity, etc.). Reissuing keys should not be done so often that it becomes burdensome, however it should be performed often enough to minimize the chance of key compromise.

- *Archive user keys for a sufficiently long cryptoperiod.*

A cryptoperiod is the time during which a key can be used for signature verification or decryption; it should extend well beyond the lifetime of a key (where the lifetime is the time during which a key can be used to generate a signature and/or perform encryption). Keys should be archived for a lengthy cryptoperiod (on the order of decades, perhaps), so that they can be used to verify signatures and decrypt ciphertext at any point during that time.

- *Handle the deactivation/revocation of keys so that data signed prior to a compromise date (or date of loss) can be verified.*

  It should be possible to designate a signing key as LOST or COMPROMISED, so signatures generated prior to a specified date can be verified. Otherwise, all data previously signed with a lost/compromised key would have to be reviewed and re-signed.

2.6 Signature Generation and Verification; Encryption and Decryption

- *Protect data prior to signature generation/verification and encryption/decryption. Be careful of how data is handled during these processes!*

  Implementors should be very careful about how data is handled before it is signed/verified (encrypted/decrypted). If the data is stored on the computer where the cryptographic function is performed, this might not pose a problem. However, if data is stored in a central database and transferred to the computer only at the time the cryptographic function is to be performed, the data should be very carefully protected during transmission. If data is not carefully protected, then an intruder could potentially alter data before a signature is generated, without the signer's knowledge.

- *Before generating a signature, users should be able to view all data to be signed. It should be made obvious to users as to exactly what data a cryptographic function is applied to.*

  User should be able to see all the data that is being signed, and it should be clearly marked for the signer. It is not always intuitive for a user to discern which data is included in a signature. Knowing what is encrypted is important, too - a user may be concerned if he knows that certain data is not being encrypted. It is not essential that all data being signed/encrypted should appear on one screen, but the user should at least be able to view all of the data before performing the cryptographic function.

- *Plan for the need of a user to re-sign data, in a tightly-controlled manner that is logged.*

  Signature verification or may fail due to a change in an organizational code, a form number, a person's last name, etc. These values might be more likely to change between signature generation and verification if they are pulled from a database to reconstruct a message. Strict controls should be put in place to restrict the use of the re-signing capability to specific situations and/or specific individuals (e.g., the original signer or a

database administrator acting on the original signer's behalf). The re-signing tool should allow a person to 1) examine what changed in the message content from the time of the original signature, and 2) decide whether or not the change warrants the generation of a new signature. All use of the re-signing tool should be carefully controlled and audited. Such an audit trail should minimally include: suspected cause of verification failure, whether or not the data was re-signed, who determined the data should be re-signed, who performed the re-signing, and the date/time of re-signing.

- *Determine what data fields must be protected using a cryptographic function.*

  The implementor should be aware of what fields are being signed and encrypted. It may not be necessary for all fields in a form to be signed and/or encrypted. Limiting the data input to a cryptographic function may have a significant impact on the speed with which that function can be performed. Fields containing sensitive data should be identified, and then a determination should be made of what cryptographic functions should be applied those fields: integrity, authenticity, and/or confidentiality.

3. Summary

The recommendations mentioned in section 2 of this document should be taken into consideration when cryptographic services for a system are being designed. However, they do not form a comprehensive list of issues that must be addressed. It is important to remember that adding cryptography to a system will not necessarily provide adequate security. Cryptography must be designed as an integrated part of the system, rather than as an add-on feature. The agency whose system was reviewed by NIST took the approach of designing cryptography into the system from the very beginning. For those situations where this cannot be done, cryptographic functions should be carefully added so that the security that they are intended to provide is not compromised.

4. References

[1] ANSI X9.17-1985, *Financial Institution Key Management (Wholesale)*, American Banker's Association, Approved April 4, 1985, Reaffirmed 1991.

[2] FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*, US DOC/NIST, January 11, 1994.

[3] FIPS PUB 46-2, *Data Encryption Standard (DES)*, US DOC/NIST, Reaffirmed December 30, 1993.

[4] FIPS PUB 186, *Digital Signature Standard (DSS)*, US DOC/NIST, May 19, 1994.

[5] FIPS PUB 180-1, *Secure Hash Standard (SHS)*, US DOC/NIST, April 17, 1995.